

Verificación de contratos y reglas en el fabric de ACI

Contenido

[Introducción](#)

[Topología](#)

[Descripción general del proceso](#)

[Identificar la regla de división en zonas/contratos utilizada](#)

[Verificar la programación del hardware](#)

[Solucionar problemas de programación de hardware](#)

[Comandos útiles de resolución de problemas](#)

[Consejos de Troubleshooting](#)

[Derivar nombre de contrato de ID de regla](#)

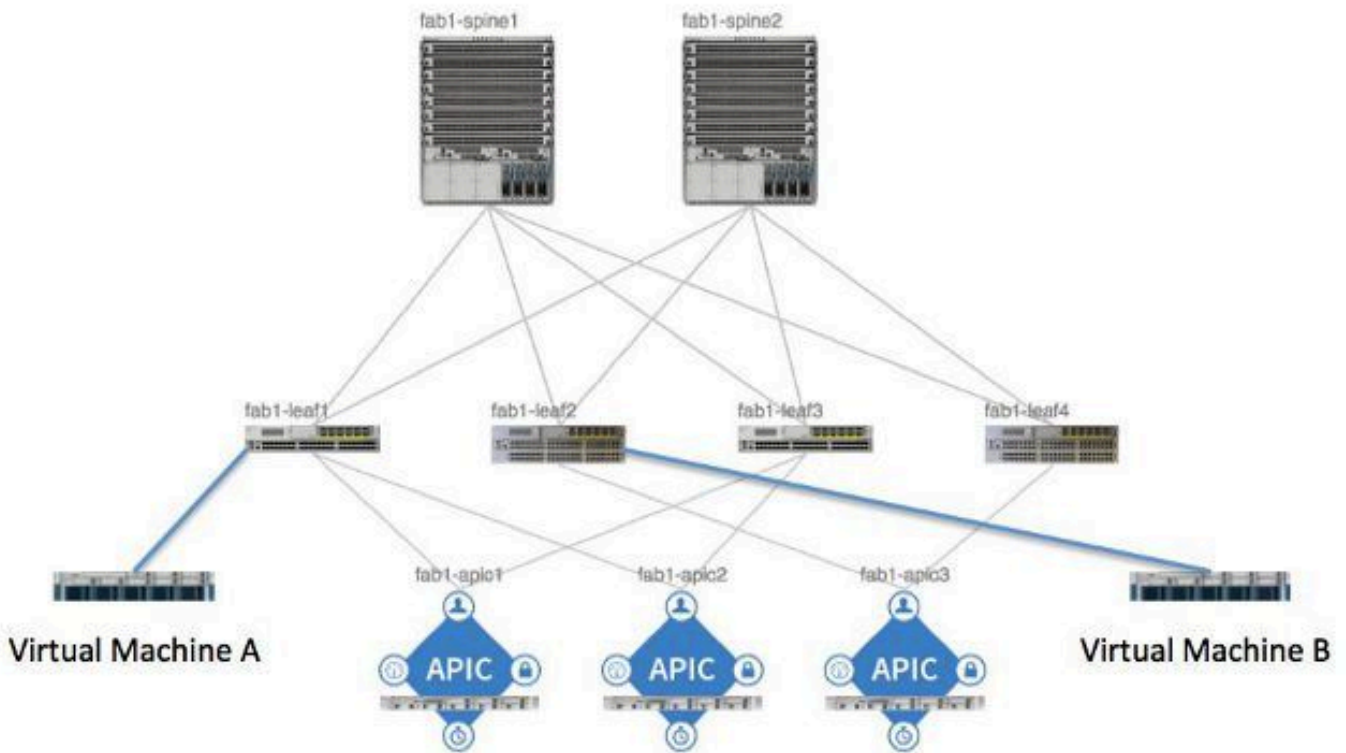
Introducción

Este documento describe cómo verificar que los contratos están configurados y se comportan correctamente en el fabric de la Infraestructura centrada en aplicaciones (ACI).

Topología

En el ejemplo que se utiliza en este documento, la máquina virtual A (VM) se adjunta a Leaf1 y se ha establecido un contrato que le permite comunicarse con VM-B, que se adjunta a Leaf2. El contrato permite el protocolo de mensajes de control de Internet (ICMP) y HTTP.

Esta imagen ilustra la topología:



Descripción general del proceso

Se trata de la interacción y el flujo de políticas para los contratos y las reglas:

1. El administrador de políticas del Application Policy Infrastructure Controller (APIC) se comunica con el administrador de elementos de políticas del switch.
2. El Administrador de elementos de política del switch programa el Almacén de objetos en el switch.
3. El administrador de políticas del switch se comunica con el cliente de calidad de servicio (ACLQOS) de la lista de control de acceso del switch.
4. El cliente ACLQOS programa el hardware.

Identificar la regla de división en zonas/contratos utilizada

A continuación se muestra un ejemplo de la salida del comando show zoning-rule de la hoja, antes de agregar el contrato para los dos grupos de punto final (EPG).

```
<#root>
fab1_leaf1#
show zoning-rule
```

Rule ID	SrcEPG	DstEPG	FilterID	operSt	Scope	Action
4096	0	0	implicit	enabled	16777200	deny,log
4097	0	0	implicit	enabled	3080192	deny,log
4098	0	0	implicit	enabled	2686976	deny,log
4099	0	49154	implicit	enabled	2686976	permit
4102	0	0	implicit	enabled	2097152	deny,log
4103	0	32771	implicit	enabled	2097152	permit
4117	16387	16386	12	enabled	2097152	permit
4116	16386	16387	13	enabled	2097152	permit
4100	16386	49154	default	enabled	2097152	permit
4101	49154	16386	default	enabled	2097152	permit
4104	0	32770	implicit	enabled	2097152	permit
4105	49155	16387	13	enabled	2097152	permit
4112	16387	49155	13	enabled	2097152	permit
4113	49155	16387	12	enabled	2097152	permit
4114	16387	49155	12	enabled	2097152	permit

[snip]

Este es el mismo resultado de comando después de agregar el contrato para que los dos EPG puedan comunicarse entre sí:

```
<#root>
```

```
fab1_leaf1#
```

```
show zoning-rule
```

Rule ID	SrcEPG	DstEPG	FilterID	operSt	Scope	Action
4096	0	0	implicit	enabled	16777200	deny,log
4097	0	0	implicit	enabled	3080192	deny,log
4098	0	0	implicit	enabled	2686976	deny,log
4099	0	49154	implicit	enabled	2686976	permit

4131	49155	32771	7	enabled	2686976	permit
4132	32771	49155	6	enabled	2686976	permit
4102	0	0	implicit	enabled	2097152	deny,log
4103	0	32771	implicit	enabled	2097152	permit
4117	16387	16386	12	enabled	2097152	permit
4116	16386	16387	13	enabled	2097152	permit
4100	16386	49154	default	enabled	2097152	permit
4101	49154	16386	default	enabled	2097152	permit
4104	0	32770	implicit	enabled	2097152	permit
4105	49155	16387	13	enabled	2097152	permit
4112	16387	49155	13	enabled	2097152	permit
4113	49155	16387	12	enabled	2097152	permit
4114	16387	49155	12	enabled	2097152	permit

[snip]

Nota: Observe los nuevos identificadores de regla (4131 y 4132) que se agregaron, los identificadores de filtro 7 y 6 y el alcance de 2686976.

Precaución: este resultado de comando le permite localizar fácilmente las reglas que debe examinar en un sistema de laboratorio; sin embargo, esto puede ser engorroso en un entorno de producción con los cambios dinámicos que se producen.

Otro método que puede utilizar para localizar las reglas de interés es utilizar Visore. Realice una búsqueda de fvCtx en el objeto administrado (MO) de contexto. A continuación, puede buscar en esa pantalla su nombre distinguido (DN) de contexto específico, como se muestra a continuación:

APIC Object Store Browser

Filter: 0 of 0

Class or DN: Filter

Property: Op: Val1: Val2:

Display URI of last query

Display last response

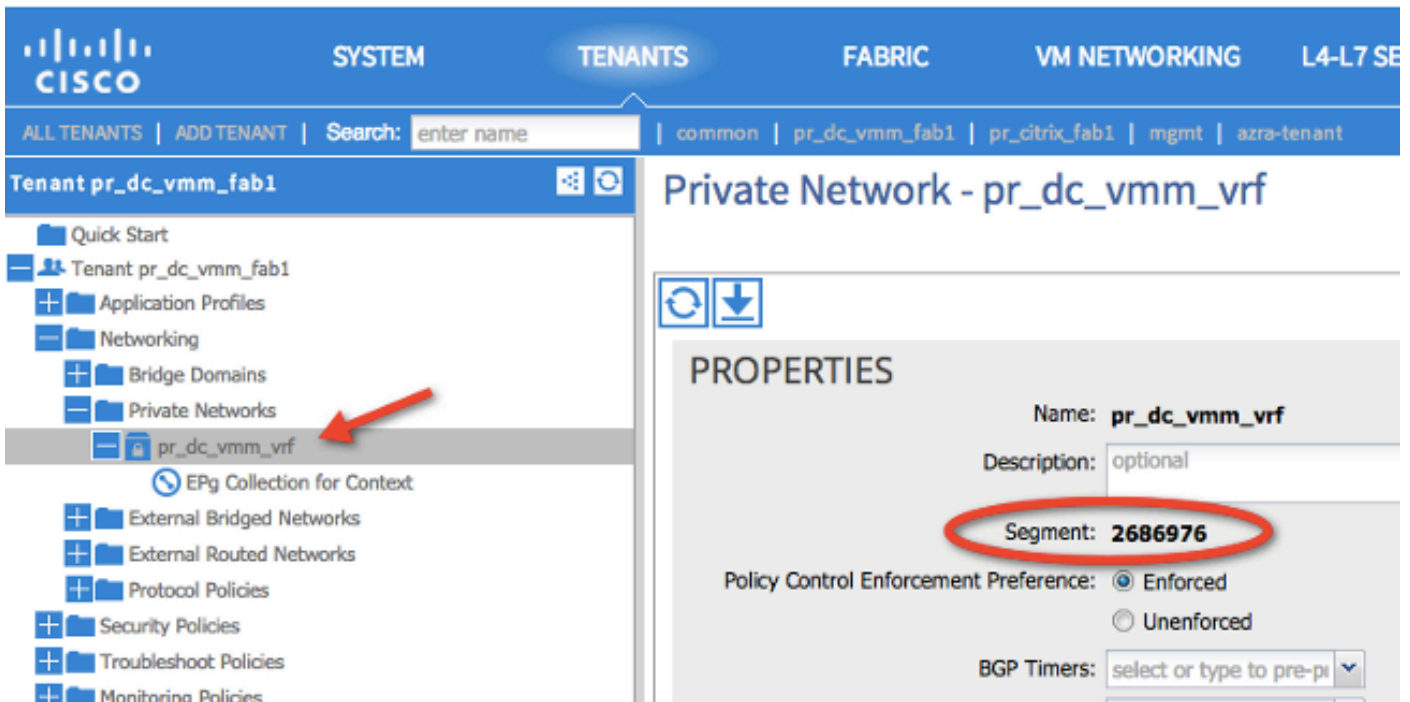
Total objects shown: 8

fvCtx		2
childAction		
descr		
dn	uni/tn-infra/ctx-overlay-1 < >	

Tomar nota del alcance de ese contexto. Puede utilizar esto para mapear al resultado del comando show-zoning-rule de modo que pueda localizar las reglas que debe consultar:

fvCtx		2
childAction		
descr		
dn	uni/tn-pr_dc_vmm_fab1/ctx-pr_dc_vmm_vrf < >	
knwMcastAct	permit	
lcOwn	local	
modTs	2014-09-03T09:32:36.625-04:00	
monPolDn	uni/tn-common/monepg-default < >	
name	pr_dc_vmm_vrf	
ownerKey		
ownerTag		
pcEntPref	enforced	
pcTag	32770	
scope	2686976	
seg	2686976	
status		
uid	15374	

También puede identificar el ID/alcance del segmento para el contexto desde la interfaz de usuario (IU), como se muestra aquí:



Este alcance coincide con el que se muestra en el resultado del comando show zoning-rules:

4098	0	0	implicit	enabled	2686976	deny, log
4099	0	49154	implicit	enabled	2686976	permit
4131	49155	32771	7	enabled	2686976	permit
4132	32771	49155	6	enabled	2686976	permit

Una vez que tenga la información de ID de alcance e identifique los ID de regla y filtro, puede utilizar el siguiente comando para verificar que se alcanzan los nuevos filtros (y no los mensajes de negación implícitos entre los EPG). Se incluye el mensaje de denegación implícito para que, de forma predeterminada, los EPG no puedan comunicarse.

Observe en este resultado de comando que Leaf1, Filter-6 (f-6) está incrementando:

```
<#root>
```

```
fab1_leaf1#
```

```
show system internal policy-mgr stats | grep 2686976
```

```
Rule (4098) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-any-f-implicit)
  Ingress: 0, Egress: 81553
```

```
Rule (4099) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-49154-f-implicit)
  Ingress: 0, Egress: 0
```

```
Rule (4131) DN (sys/actrl/scope-2686976/rule-2686976-s-49155-d-32771-f-7)
  Ingress: 0, Egress: 0
```

```
Rule (4132) DN (sys/actrl/scope-2686976/rule-2686976-s-32771-d-49155-f-6)
  Ingress: 1440, Egress: 0
```

```
<#root>
```

```
fab1_leaf1#
```

```
show system internal policy-mgr stats | grep 2686976
```

```
Rule (4098) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-any-f-implicit)
```

```
Ingress: 0, Egress: 81553
```

```
Rule (4099) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-49154-f-implicit)
```

```
Ingress: 0, Egress: 0
```

```
Rule (4131) DN (sys/actrl/scope-2686976/rule-2686976-s-49155-d-32771-f-7)
```

```
Ingress: 0, Egress: 0
```

```
Rule (4132) DN (sys/actrl/scope-2686976/rule-2686976-s-32771-d-49155-f-6)
```

```
Ingress: 1470, Egress: 0
```

Observe en este resultado de comando que Leaf2, Filter-7 (f-7) está incrementando:

```
<#root>
```

```
fab1_leaf2#
```

```
show system internal policy-mgr stats | grep 268697
```

```
Rule (4098) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-any-f-implicit)
```

```
Ingress: 0, Egress: 80257
```

```
Rule (4099) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-49153-f-implicit)
```

```
Ingress: 0, Egress: 0
```

```
Rule (4117) DN (sys/actrl/scope-2686976/rule-2686976-s-32771-d-49155-f-6)
```

```
Ingress: 0, Egress: 0
```

```
Rule (4118) DN (sys/actrl/scope-2686976/rule-2686976-s-49155-d-32771-f-7)
```

```
Ingress: 2481, Egress: 0
```

```
<#root>
```

```
fab1_leaf2#
```

```
show system internal policy-mgr stats | grep 268697
```

```
Rule (4098) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-any-f-implicit)
```

```
Ingress: 0, Egress: 80257
```

Rule (4099) DN (sys/actrl/scope-2686976/rule-2686976-s-any-d-49153-f-implicit)
Ingress: 0, Egress: 0

Rule (4117) DN (sys/actrl/scope-2686976/rule-2686976-s-32771-d-49155-f-6)
Ingress: 0, Egress: 0

Rule (4118) DN (sys/actrl/scope-2686976/rule-2686976-s-49155-d-32771-f-7)
Ingress: 2511, Egress: 0

Sugerencia: el conocimiento del alcance, ID de regla, destino, pcTags de origen y filtro es importante cuando se intenta solucionar este problema con mayor profundidad. También resulta útil conocer los EPG entre los que existe el ID de regla.

Puede realizar una búsqueda en el MO con el nombre DN fvAEPg y grep para la pcTag particular mediante el comando moquery, como se muestra aquí:

```
<#root>
admin@RTP_Apic1:~>
moquery -c fvAEPg | grep 49155 -B 5

dn : uni/tn-Prod/ap-commercespace/
epg-Web

lcOwn : local
matchT : AtleastOne
modTs : 2014-10-16T01:27:35.355-04:00
monPolDn : uni/tn-common/monepg-default

pcTag : 49155
```

También puede utilizar la opción filter con el comando moquery, como se muestra aquí:

```
<#root>
admin@RTP_Apic1:~>
moquery -c fvAEPg -f 'fv.AEPg.pcTag=="49155"'

Total Objects shown: 1

# fv.AEPg
name : Web
childAction :
configIssues :
configSt : applied
descr :
```


dn : uni/tn-Prod/ap-commerceworkspace/

epg-Web

lcOwn : local

matchT : AtleastOne

modTs : 2014-10-16T01:27:35.355-04:00

monPolDn : uni/tn-common/monepg-default

pcTag : 49155

prio : unspecified

rn : epg-Web

scope : 2523136

status :

triggerSt : triggerable

uid : 15374

Verificar la programación del hardware

Ahora puede verificar la entrada de hardware para la regla. Para ver la información de hardware, ingrese el comando `show platform internal ns table`

`mth_lux_slvz_DHS_SecurityGroupStatTable_memif_data ingress` (este es un comando `vsh_lc`):

```
module-1# show platform internal ns table mth_lux_slvz_DHS_SecurityGroupStatTable_memif_data ingress
error opening file
: No such file or direct
Last login: Fri Sep  5 1
=====
[Restored]
TABLE INSTANCE : 0
=====
ENTRY[000010] = pkt_cnt=0x5176e
ENTRY[000011] = pkt_cnt=0x7d95
ENTRY[000014] = pkt_cnt=0x9d414
ENTRY[000016] = pkt_cnt=0x15288a
ENTRY[000017] = pkt_cnt=0x2975ce
ENTRY[000018] = pkt_cnt=0x662b
ENTRY[000021] = pkt_cnt=0x329f
ENTRY[000023] = pkt_cnt=0x40
ENTRY[000024] = pkt_cnt=0x21bf
ENTRY[000026] = pkt_cnt=0x556f0
ENTRY[000029] = pkt_cnt=0x5d7e2
ENTRY[000041] = pkt_cnt=0x6360
ENTRY[000050] = pkt_cnt=0x2a05
ENTRY[000052] = pkt_cnt=0x5ec
ENTRY[000054] = pkt_cnt=0xdfd
ENTRY[000055] = pkt_cnt=0xd
ENTRY[000068] = pkt_cnt=0xdac
ENTRY[000072] = pkt_cnt=0x91
ENTRY[000077] = pkt_cnt=0x35b
module-1# show platform internal ns table mth_lux_slvz_DHS_SecurityGroupStatTable_memif_data ingress
error opening file
: No such file or directory
=====
TABLE INSTANCE : 0
=====
ENTRY[000010] = pkt_cnt=0x517cf
ENTRY[000011] = pkt_cnt=0x7d9f
ENTRY[000014] = pkt_cnt=0x9d494
ENTRY[000016] = pkt_cnt=0x152262
ENTRY[000017] = pkt_cnt=0x29799e5
ENTRY[000018] = pkt_cnt=0x6631
ENTRY[000021] = pkt_cnt=0x329f
ENTRY[000023] = pkt_cnt=0x40
ENTRY[000024] = pkt_cnt=0x21c6
ENTRY[000026] = pkt_cnt=0x55771
ENTRY[000029] = pkt_cnt=0x5d7e2
ENTRY[000041] = pkt_cnt=0x64e0
ENTRY[000050] = pkt_cnt=0x2a05
ENTRY[000052] = pkt_cnt=0x5ec
ENTRY[000054] = pkt_cnt=0xdfd
ENTRY[000055] = pkt_cnt=0xd
ENTRY[000068] = pkt_cnt=0xdb8
ENTRY[000072] = pkt_cnt=0x92
ENTRY[000077] = pkt_cnt=0x35b
```

En este ejemplo, la entrada de hardware 41 (ENTRY [000041]) está aumentando.

Nota: El comando anterior que se muestra se utiliza para el ASIC Northstar. El comando que se utiliza para Donner o Donner+ es show platform internal ns table mth_luxh_slvy_DHS_SecurityGroupStatTable_memif_data.

Nota: El uso de este comando no es práctico en un entorno de producción, pero puede

utilizar en su lugar los otros comandos que se describen en esta sección.

Recuerde la regla (4132) y el alcance (268976).

4098	0	0	implicit	enabled	2686976	deny,log
4099	0	49154	implicit	enabled	2686976	permit
4131	49155	32771	7	enabled	2686976	permit
4132	32771	49155	6	enabled	2686976	permit

Ingrese este comando para determinar el ID de regla para la asignación de entrada de índice de hardware de Memoria Direccional por Contenido Ternario (TCAM) y filtre según el ID de regla y/o el ID de filtro:

```
<#root>
```

```
module-1#
```

```
show system internal aclqos zoning-rules
```

```
[snip]
```

```
=====  
Rule ID: 4131 Scope 4 Src EPG: 49155 Dst EPG: 32771 Filter 7
```

```
Curr TCAM resource:
```

```
=====  
unit_id: 0  
=== Region priority: 771 (rule prio: 3 entry: 3)===  
sw_index = 62 |
```

```
hw_index = 40
```

```
=== Region priority: 772 (rule prio: 3 entry: 4)===  
sw_index = 63 |
```

```
hw_index = 45
```

```
=====  
Rule ID: 4132 Scope 4 Src EPG: 32771 Dst EPG: 49155 Filter 6
```

```
Curr TCAM resource:
```

```
=====  
unit_id: 0  
=== Region priority: 771 (rule prio: 3 entry: 3)===  
sw_index = 66 |
```

```
hw_index = 41
```

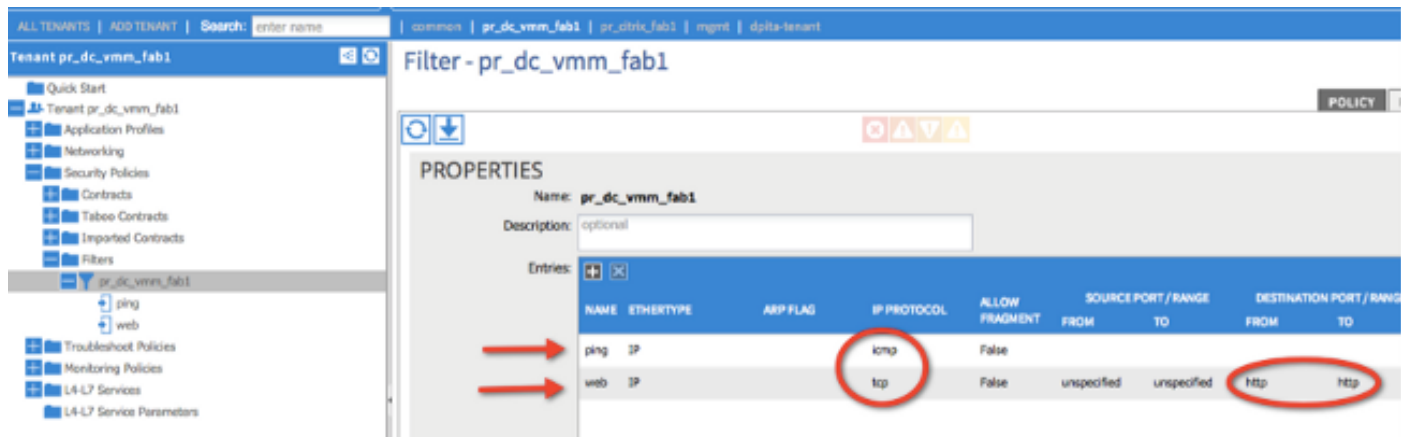
```
=== Region priority: 771 (rule prio: 3 entry: 3)===  
sw_index = 67 |
```

```
hw_index = 42
```

[snip]

Para este ejemplo, la combinación de interés de EPG de origen y destino es 32771=0x8003, 49155=0xC003. Por lo tanto, puede considerar todas las entradas TCAM para estas clases de origen y destino que coincidan con los ID de regla (4131 y 4132) y los ID de filtro (6 y 7).

En este ejemplo, algunas de estas entradas TCAM se descartan. A modo de referencia, a continuación se muestra la configuración del contrato que permite pings y tráfico web para estos EPG:



<#root>

module-1#

show platform internal ns table mth_lux_slvz_DHS_SecurityGroupKeyTable0

_memif_data 41

```

=====
TABLE INSTANCE : 0
=====
ENTRY[000041] =
    sg_label=0x4
    sclass=0x8003
    dclass=0xc003
    prot=0x1
(IP Protocol 0x01 = ICMP)

```

Nota: El comando anterior que se muestra se utiliza para el ASIC Northstar. El comando que

se utiliza para Donner o Donner+ es show platform internal ns table mth_luxh_slvq_DHS_SecurityGroupKeyTable0_memif_data.

Decimal	Keyword	Protocol	IPv6 Extension Header	
0	HOPOPT	IPv6 Hop-by-Hop Option	Y	[RFC2460]
1	ICMP	Internet Control Message		[RFC792]
2	IGMP	Internet Group Management		[RFC1112]

<#root>

```
sup_tx_mask=0x1
  src_policy_incomplete_mask=0x1
  dst_policy_incomplete_mask=0x1
  class_eq_mask=0x1
  aclass_mask=0x1fff
  port_dir_mask=0x1
  dport_mask=0xffff
  sport_mask=0xffff
  tcpflags_mask=0xff
  ip_opt_mask=0x1
  ipv6_route_mask=0x1
  ip_fragment_mask=0x1
  ip_frag_offset0_mask=0x1
  ip_frag_offset1_mask=0x1
  ip_mf_mask=0x1
  t4_partial_mask=0x1
  dst_local_mask=0x1
  routeable_mask=0x1
  spare_mask=0x7fff
  v4addr_key_mask=0x1
  v6addr_key_mask=0x1
  valid=0x1
```

module-1#

show platform internal ns table mth_lux_slvz_DHS_SecurityGroupKeyTable0

_memif_data 42

=====

TABLE INSTANCE : 0

=====

ENTRY[000042] =

sg_label=0x4

sclass=0x8003

dc1ass=0xc003

prot=0x6

<--

dport=0x50

<--

Decimal ☒	Keyword ☒	Protocol ☒	IPv6 Extension Header ☒	
0	HOPOPT	IPv6 Hop-by-Hop Option	Y	[RFC2460]
1	ICMP	Internet Control Message		[RFC792]
2	IGMP	Internet Group Management		[RFC1112]
3	GGP	Gateway-to-Gateway		[RFC823]
4	IPv4	IPv4 encapsulation		[RFC2003]
5	ST	Stream		[RFC1190] [RFC1819]
6	TCP	Transmission Control		[RFC793]
7	CBT	CBT		[Tony Ballardie]

Port ↕	TCP ↕	UDP ▲	Description
0	TCP		Programming technique for specifying system-allocated (dynamic) ports ^[3]
21	TCP		FTP control (command)
25	TCP		Simple Mail Transfer Protocol (SMTP)—used for e-mail routing between mail servers
43	TCP		WHOIS protocol
57	TCP		Mail Transfer Protocol (RFC 780 ↗)
70	TCP		Gopher protocol
71	TCP		NETRJS protocol
72	TCP		NETRJS protocol
73	TCP		NETRJS protocol
74	TCP		NETRJS protocol
79	TCP		Finger protocol
80	TCP		Hypertext Transfer Protocol (HTTP) ^[12]
81	TCP		Terminology: Onion routing

sup_tx_mask=0x1

src_policy_incomplete_mask=0x1

dst_policy_incomplete_mask=0x1

class_eq_mask=0x1

aclass_mask=0x1ff

port_dir_mask=0x1

sport_mask=0xffff

tcpflags_mask=0xff

ip_opt_mask=0x1

ipv6_route_mask=0x1

ip_fragment_mask=0x1

ip_frag_offset0_mask=0x1

ip_frag_offset1_mask=0x1

ip_mf_mask=0x1

l4_partial_mask=0x1

dst_local_mask=0x1

Sugerencia: puede verificar cada una de las entradas TCAM con el mismo método.

Solucionar problemas de programación de hardware

Esta sección proporciona algunos comandos y sugerencias útiles para la solución de problemas.

Comandos útiles de resolución de problemas

Estos son algunos comandos útiles que puede utilizar para localizar los errores de Policy Manager de hoja cuando se encuentran problemas:

```
<#root>
```

```
fab1_leaf1#
```

```
show system internal policy-mgr event-history errors
```

```
1) Event:E_DEBUG, length:84, at 6132 usecs after Mon Sep 8 13:15:56 2014
```

```
[103] policy_mgr_handle_ctx_mrules(779): ERROR: Failed to process prio(1537):  
(null)
```

```
2) Event:E_DEBUG, length:141, at 6105 usecs after Mon Sep 8 13:15:56 2014
```

```
[103] policy_mgr_process_mrule_prio_aces(646): ERROR: Failed to insert iptables  
rule for rule(4120) , fentry(5_0) with priority(1537): (null)
```

```
[snip]
```

```
fab1_leaf1#
```

```
show system internal policy-mgr event-histor trace
```

```
[1409945922.23737] policy_mgr_ppf_hdl_close_state:562: Got close state callback
```

```
[1409945922.23696] policy_mgr_ppf_rdy_ntf_fun:239: StatStoreEnd returned: 0x0(SU  
CCESS)
```

```
[1409945922.23502] policy_mgr_ppf_rdy_ntf_fun:208: ppf ready notification: sess_  
id: (0xFF0104B400005B51)
```

```
[1409945922.23475] policy_mgr_ppf_rdy_ntf_fun:205: Got ready notification callba  
ck with statustype (4)
```

```
[1409945921.983476] policy_mgr_gwrap_handler:992: Dropped...now purging it...
```


[1409945921.982882] policy_mgr_ppf_goto_state_fun:481: Sess id (0xFF0104B400005B

[snip]

module-1#

show system internal aclqos event-history trace

T [Fri Sep 5 13:18:24.863283] ===== Session End =====

T [Fri Sep 5 13:18:24.862924] Commit phase: Time taken 0.62 ms, usr 0.00 ms,
sys 0.00 ms

T [Fri Sep 5 13:18:24.862302] ppf session [0xff0104b410000087] commit ... npi
nst 1

T [Fri Sep 5 13:18:24.861421] Verify phase: Time taken 0.77 ms, usr 0.00 ms,
sys 0.00 ms

T [Fri Sep 5 13:18:24.860615] ===== Session Begin =====

T [Fri Sep 5 13:18:24.830472] ===== Session End =====

T [Fri Sep 5 13:18:24.830062] Commit phase: Time taken 0.98 ms, usr 0.00 ms,
sys 0.00 ms

T [Fri Sep 5 13:18:24.829085] ppf session [0xff0104b410000086] commit ... npi
nst 1

T [Fri Sep 5 13:18:24.827685] Verify phase: Time taken 2.04 ms, usr 0.00 ms,
sys 0.00 ms

T [Fri Sep 5 13:18:24.825388] ===== Session Begin =====

T [Fri Sep 5 12:32:51.364225] ===== Session End =====

T [Fri Sep 5 12:32:51.363748] Commit phase: Time taken 0.64 ms, usr 0.00 ms,

[snip]

Sugerencia: algunos de los archivos son grandes, por lo que es más fácil enviarlos a la memoria flash de inicialización y examinarlos en un editor.

<#root>

module-1#

show system internal aclqos ?

asic Asic information
brcm Broadcam information
database Database
event-history Show various event logs of ACLQOS
mem-stats Show memory allocation statistics of ACLQOS
prefix External EPG prefixes
qos QoS related information
range-resource Zoning rules L4 destination port range resources
regions Security TCAM priority regions
span SPAN related information
zoning-rules Show zoning rules

module-1#

show system internal aclqos event-history ?

errors Show error logs of ACLQOS

msgs Show various message logs of ACLQOS
ppf Show ppf logs of ACLQOS
ppf-parse Show ppf-parse logs of ACLQOS
prefix Show prefix logs of ACLQOS
qos Show qos logs of ACLQOS
qos-detail Show detailed qos logs of ACLQOS
span Show span logs of ACLQOS
span-detail Show detailed span logs of ACLQOS

trace Show trace logs of ACLQOS

trace-detail Show detailed trace logs of ACLQOS

zoning-rules Show detailed logs of ACLQOS

Consejos de Troubleshooting

Estos son algunos consejos útiles para la resolución de problemas:

- Si parece que experimenta un problema de agotamiento de TCAM, verifique la UI o la CLI para detectar fallas asociadas con la regla en cuestión. Este fallo puede notificarse:

```
<#root>
```

```
Fault F1203 - Rule failed due to hardware programming error.
```

Una regla puede aceptar más de una entrada TCAM en el circuito integrado específico de la aplicación (ASIC). Para ver el número de entradas en el ASIC, ingrese estos comandos:

```
<#root>
```

```
fab1-leaf1#
```

```
vsh_lc
```

```
module-1#
```

```
show platform internal ns table-health
```

```
VLAN STATE curr usage: 0 - size: 4096  
QQ curr usage: 0 - size: 16384  
SEG STATE curr usage: 0 - size: 4096  
SRC TEP curr usage: 0 - size: 4096  
POLICY KEY curr usage: 0 - size: 1  
SRC VP curr usage: 0 - size: 4096
```

```
SEC GRP curr usage: 43 - size: 4096
```

Nota: En este ejemplo, hay 43 entradas presentes. Este uso también se informa al APIC en la clase eqptCapacity.

- Cuando hay varias coincidencias, la búsqueda TCAM devuelve el hw-index inferior. Para verificar el índice, ingrese este comando:

```
<#root>
```

```
show system internal aclqos zoning-rule
```

Al solucionar problemas, puede observar la caída causada por la regla implícita any-any.

Esta regla siempre está en la parte inferior, lo que significa que el paquete se descarta porque no existe una regla. Esto se debe a un error de configuración o el Administrador de elementos de política no lo programa como se esperaba.

- pcTags puede tener un alcance local o global:

System Reserved pcTag - Esta pcTag se utiliza para las reglas internas del sistema (1-15).

Globally scoped pcTag - Esta pcTag se utiliza para el servicio compartido (16-16385).

pcTag con ámbito local: este pcTag se utiliza localmente por VRF (intervalo de 16386-65535).

Al resolver problemas, un vistazo rápido a la longitud del valor indica su alcance.

Derivar nombre de contrato de ID de regla

A menudo, cuando se encuentra en un caso de resolución de problemas, un ingeniero está estudiando las reglas de zonificación. En algunos casos, un EPG/pcTag tiene muchos contratos y puede ser engorroso resolver problemas. Esta sección trata sobre una manera de determinar el nombre del contrato que se está utilizando entre los EPG/pcTags a partir del ID de regla que se ve en la CLI del switch.

Para comenzar, consulte el objeto de contrato/regla concreto `actrlRule` si lo desea, restrinja la búsqueda por propiedad: `id valor: rule-d`

Una vez encontrada la regla correcta, haga clic en la flecha verde del DN para ver los objetos `actrlRule` secundarios. Los niños es donde está nuestra respuesta.

actrlRule ?	
action	permit
actrlCfgFailedBmp	
actrlCfgFailedTs	00:00:00:00.000
actrlCfgState	0
childAction	
dPcTag	16388 ←
descr	
direction	uni-dir
dn	topology/pod-1/node-101/sys/actrl/scope-2719746/rule-2719746-s-49164-d-16388-f-38 < > 📊 ! 🛡️
fltId	38 ←
id	4143 ←
lcOwn	local
markDscp	unspecified
modTs	2016-01-08T19:44:02.267+00:00
monPolDn	uni/tn-common/monepg-default < > 📊 ! 🛡️
name	
operSt	enabled
operStQual	
prio	fully_qual
qosGrp	unspecified
sPcTag	49164 ←
scopeId	2719746 ←
status	
type	tenant

El objeto secundario aquí es actrlRsToEpgConn. Normalmente puede haber dos, uno para cada EPG. El DN de este objeto muestra los dos EPG entre los que se aplica el contrato, así como la dirección (proveedor o consumidor) y, lo que es más importante, el nombre del objeto del contrato.

actrlRsToEpgConn	
childAction	
dn	topology/pod-1/node-101/sys/actrl/scope-2719746/rule-2719746-s-49164-d-16388-f-38/rstoEpgConn-[cdef-[uni/tn-dpita-tenant/brc-dpita-ssh]/epgCont-[uni/tn-dpita-tenant/ap-dpita-AP/epg-dpita-EPG1]/fr-[uni/tn-dpita-tenant/brc-dpita-ssh/dirass-prov-[uni/tn-dpita-tenant/ap-dpita-AP/epg-dpita-EPG1]-any-no]/to-[uni/tn-dpita-tenant/brc-dpita-ssh/dirass-cons-[uni/tn-dpita-tenant/ap-dpita-AP/epg-dpita-EPG2]-any-no]] < >   
forceResolve	no
leOwn	local
modTs	2016-01-08T19:44:02.267+00:00
rType	mo
state	unformed
stateQual	none
status	
tCl	vzToEPg
tDn	cdef-[uni/tn-dpita-tenant/brc-dpita-ssh]/epgCont-[uni/tn-dpita-tenant/ap-dpita-AP/epg-dpita-EPG1]/fr-[uni/tn-dpita-tenant/brc-dpita-ssh/dirass-prov-[uni/tn-dpita-tenant/ap-dpita-AP/epg-dpita-EPG1]-any-no]/to-[uni/tn-dpita-tenant/brc-dpita-ssh/dirass-cons-[uni/tn-dpita-tenant/ap-dpita-AP/epg-dpita-EPG2]-any-no]] < >   
tType	mo

Como se ha resaltado, el nombre del contrato en este caso es brc-dpita-ssh.

Si es necesario, consulte vzBrCP para encontrar el contrato correcto.

vzBrCP

?

childAction	
configIssues	
descr	
dn	uni/tn-dpita-tenant/brc-dpita-ssh < > ! H
lcOwn	local
modTs	2015-06-25T16:21:10.003+00:00
monPolDn	uni/tn-common/monepg-default < > ! H
name	dpita-ssh
ownerKey	
ownerTag	
prio	unspecified
reevaluateAll	no
scope	context
status	
uid	15374

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).